

On Throughput-Smoothness Trade-offs in Streaming Communication

1

Gauri Joshi, *Student Member, IEEE*, Yuval Kochman, *Member, IEEE*, Gregory W. Wornell, *Fellow, IEEE*

Abstract

Unlike traditional file transfer where only total delay matters, streaming applications impose delay constraints on each packet and require them to be in order. To achieve fast in-order packet decoding, we have to compromise on the throughput. We study this trade-off between throughput and smoothness in packet decoding. We first consider a point-to-point streaming and analyze how the trade-off is affected by the frequency of block-wise feedback, whereby the source receives full channel state feedback at periodic intervals. We show that frequent feedback can drastically improve the throughput-smoothness trade-off. Then we consider the problem of multicasting a packet stream to two users. For both point-to-point and multicast streaming, we propose a spectrum of coding schemes that span different throughput-smoothness tradeoffs. One can choose an appropriate coding scheme from these, depending upon the delay-sensitivity and bandwidth limitations of the application. This work introduces a novel style of analysis using renewal processes and Markov chains to analyze coding schemes.

Index Terms

erasure codes, streaming, in-order packet delivery, throughput-delay trade-off

This work was presented in the INFOCOM workshop on Communication and Networking Techniques for Contemporary Video, Apr 2014, and the International Symposium on Network Coding, Jun 2014.

G. Joshi and G. W. Wornell are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA. Y. Kochman is with the School of Computer Science and Engineering, the Hebrew University of Jerusalem, Israel. (E-mail: gauri@mit.edu, yuvalko@cs.huji.ac.il, gww@mit.edu)

I. INTRODUCTION

A. Motivation

A recent report [1] shows that 62% of the Internet traffic in North America comes from real-time streaming applications. Unlike traditional file transfer where only total delay matters, streaming imposes delay constraints on each individual packet. Further, many applications require in-order playback of packets at the receiver. Packets received out of order are buffered until the missing packets in the sequence are successfully decoded. In audio and video applications some packets can be dropped without affecting the streaming quality. However, other applications such as remote desktop, and collaborative tools such as Dropbox [2] and Google Docs [3] have strict order constraints on packets, where packets represent instructions that need to be executed in order at the receiver.

Thus, there is a need to develop transmission schemes that can ensure in-order packet delivery to the user, with efficient use of available bandwidth. To ensure that packets are decoded in order, the transmission scheme must give higher priority to older packets that were delayed, or received in error. However, repeating old packets instead of transmitting new packets results in a loss in the overall rate of packet delivery to the user, i.e., the throughput. Thus there is a fundamental trade-off between throughput and in-order decoding delay.

The throughput loss incurred to achieve smooth in-order packet delivery can be significantly reduced if the source receives feedback about packet losses. Then the source can adapt its future transmission strategy to strike the right balance between old and new packets. We study this interplay between feedback and the throughput-smoothness trade-off. This analysis can help design transmission schemes that achieve the best throughput-smoothness trade-off with a limited amount of feedback.

Further, if there are more than one users that request a common stream of packets, then there is an additional inter-dependence between the users. Since the users decode different sets of packets depending of their channel erasures, a packet that is innovative and in-order for one user, may be redundant for another. Thus, the source must strike a balance of giving priority to each of the users. We present a framework to analyze such multicast streaming scenarios.

B. Previous Work

When there is immediate and error-free feedback, it is well understood that a simple Automatic-repeat-request (ARQ) scheme is both throughput and delay optimal. But only a few papers in literature have analyzed streaming codes with delayed or no feedback. Fountain codes [4] are capacity-achieving erasure codes, but they are not suitable for streaming because the decoding delay is proportional to the size of the data. Streaming codes without feedback for constrained channels such as adversarial and cyclic burst erasure channels were first proposed in [5], and also extensively explored in [6], [7]. The thesis [5] also proposed codes for more general erasure models and analyzed their decoding delay. These codes are based upon sending linear combinations of source packets; indeed, it can be shown that there is no loss in restricting the codes to be linear.

However, decoding delay does not capture *in order* packet delivery, which is required for streaming applications. This aspect is captured in the delay metrics in [8] and [9], which consider that packets are played in-order at the receiver. The authors in [8] analyze the playback delay of real-time streaming for uncoded packet transmission over a channel with long feedback delay. In [9], [10] we show that the number of interruptions in playback scales $\Theta(\log n)$ for a stream of length n . We then proposed codes that minimize the pre-log term for the no feedback and immediate feedback cases. Recent work [11] shows that if the stream is available beforehand (non real-time), it is possible to guarantee constant number of interruptions, independent of n . This analysis of in-order playback delay blends tools from renewal processes and large deviations with traditional coding theory. These tools were later employed in interesting work; see e.g. [12], [13] which also consider in-order packet delivery delay. In this paper we aim to understand how the frequency of feedback about erasures affects the design of codes to ensure smooth point-to-point streaming.

When the source needs to stream the data to multiple users, even the immediate feedback case becomes non-trivial. The use of network coding in multicast packet transmission has been studied in [14]–[19]. The authors in [14] use as a delay metric the number of coded packets that are successfully received, but do not allow immediate decoding of a source packet. For two users, the paper shows that a greedy coding scheme is throughput-optimal and guarantees

immediate decoding in every slot. However, optimality of this scheme has not been proved for three or more users. In [15], the authors analyze decoding delay with the greedy coding scheme in the two user case. However, both these delay metrics do not capture the aspect of in-order packet delivery.

In-order packet delivery for multicast with immediate feedback is considered in [16]–[18]. These works consider that packets are generated by a Poisson process and are greedily added to all future coded combinations. Another related work is [19] which also considers Poisson packet generation in a two-user multicast scenario and derives the stability condition for having finite delivery delay. However, in practice the source can use feedback about past erasures to decide which packets to add to the coded combinations, instead of just greedy coding over all generated packets. In this paper we consider this model of source-controlled packet transmission.

C. Our Contributions

For the point-to-point streaming scenario, also presented in [20], we consider the problem of how to effectively utilize feedback received by the source to ensure in-order packet delivery to the user. We consider block-wise feedback, where the source receives information about past channel states at periodic intervals. In contrast to playback delay considered in [8] and [9], we propose a delay metric called the smoothness exponent. It captures the burstiness in the in-order packet delivery. In the limiting case of immediate feedback, we can use ARQ and achieve the optimal throughput and smoothness simultaneously. But as the feedback delay increases, we have to compromise on at least one of these metrics. Our analysis shows that for the same throughput, having more frequent block-wise feedback significantly improves the smoothness of packet delivery. This conclusion is reminiscent of [21] which studied the effect of feedback on error exponents. We present a spectrum of coding schemes spanning the throughput-smoothness trade-off, and prove that they give the best trade-off within a broad class of schemes for the no feedback, and small feedback delay cases.

Next, we extend this analysis to the multicast scenario. We focus of two user case with immediate feedback to the source. Since each user may decode a different set of packets depending of the channel erasures, the in-order packet required by one user may be redundant to the other user. Thus, giving priority to one user can cause a throughput loss to the other. We

analyze this interplay between the throughput-smoothness trade-offs of the two users using a Markov chain model for in-order packet decoding. This work was presented in part in [22].

D. Organization

The rest of the paper is organized as follows. In Section III we analyze the effect of block-wise feedback on the throughput-smoothness trade-off in point-to-point streaming. In Section IV we present a framework for analyzing in-order packet delivery in multicast streaming with immediate feedback. Section V summarizes the results and presents future directions. The longer proofs are deferred to the appendix.

II. PRELIMINARIES

A. Source and Channel Model

The source has a large stream of packets s_1, s_2, \dots, s_n . The encoder creates a coded packet $y_n = f(s_1, s_2 \dots s_n)$ in each slot n and transmits it over the channel. The encoding function f is known to the receiver. For example, if y_n is a linear combination of the source packets, the coefficients are included in the transmitted packet so that the receiver can use them to decode the source packets from the coded combination. Without loss of generality, we can assume that y_n is a linear combination of the source packets. The coefficients are chosen from a large enough field such that the coded combinations are independent with high probability.

Each coded combination is transmitted to K users U_1, U_2, \dots, U_K . We consider an i.i.d. erasure channel to each user such that every transmitted packet is received successfully at user U_k with probability p_k , and otherwise received in error and discarded. The erasure events are independent across the users. An erasure channel is a good model when encoded packets have a set of checksum bits that can be used to verify with high probability whether the received packet is error-free. For the single-user case considered in Section III, we denote the channel success probability as p , without the subscript.

B. Packet Delivery

The application at each user requires the stream of packets to be *in order*. Packets received out of order are buffered until the missing packets in the sequence are decoded. We assume

that the buffer is large enough to store all the out-of-order packets. Every time the earliest undecoded packet is decoded, a burst of in-order decoded packets is delivered to the application. For example, suppose that s_1 has been delivered and s_3, s_4, s_6 are decoded and waiting in the buffer. If s_2 is decoded in the next slot, then s_2, s_3 and s_4 are delivered to the application.

C. Feedback Model

We consider that the source receives block-wise feedback about channel erasures after every d slots. Thus, before transmitting in slot $kd + 1$, for all integers $k \geq 1$, the source knows about the erasures in slots $(k - 1)d + 1$ to kd . It can use this information to adapt its transmission strategy in slot $kd + 1$. Block-wise feedback can be used to model a half-duplex communication channel where after every d slots of packet transmission, the channel is reserved for each user to send d bits of feedback to the source about the status of decoding. The extreme case $d = 1$, corresponds to immediate feedback when the source has complete knowledge about past erasures. And when $d \rightarrow \infty$, the block-wise feedback model converges to the scenario where there is no feedback to the source.

Note that the feedback can be used to estimate p_i for $1 \leq i \leq K$, the success probabilities of the erasure channels, when they are unknown to the source. Thus, the coding schemes we propose for $d < \infty$ are universal; they can be used even when the channel quality of unknown to the source.

D. Notions of Packet Decoding

We now define some notions of packet decoding that aid the presentation and analysis of coding schemes in the rest of the paper.

Definition 1 (Innovative Packets). *A coded packet is said to be innovative to a user if it is linear independent with respect to the coded packets received by that user until that time.*

Definition 2 (Seen Packets). *The transmitted marks a packet s_k as “seen” by a user when it knows that the user has successfully received a coded combination that only includes s_k and packets s_i for $1 \leq i < k$.*

Since each user requires packets strictly in-order, the transmitter can stop including s_k in coded packets when it knows that all the users have seen it. This is because the users can decode s_k once all s_i for $i < k$ are decoded.

Definition 3 (Required packet). *The required packet of U_i is its earliest undecoded packet. Its index is denoted by r_i .*

In other words, s_{r_i} is the first unseen packet of user U_i . For example, if packets s_1 , s_3 and s_4 have been decoded at user U_i , its required packet s_{r_i} is s_2 .

E. Throughput and Delay Metrics

We now define the metrics for throughput and smoothness in packet delivery. We define them for a single user here, but the definitions directly extend to multiple users.

Definition 4 (Throughput). *If I_n is the number of packets delivered in-order to a user until time n , the throughput τ is,*

$$\lim_{n \rightarrow \infty} \frac{I_n}{n} \text{ in probability.} \quad (1)$$

The maximum possible throughput is $\tau = p$, where p is the success probability of our erasure channel. The receiver application may require a minimum level of throughput. For example, if applications with playback require τ to be greater than the playback rate.

The throughput captures the overall rate at which packets are delivered, irrespective of their delays. If the channel did not have any erasures, packet s_k would be delivered to the user in slot k . The random erasures, and absence of immediate feedback about past erasures results in variation in the time at which packets are delivered. We capture the burstiness in packet delivery using the following delay metric.

Definition 5 (Smoothness Exponent). *Let D_k be in-order decoding delay of packet s_k , the earliest time at which all packets p_1, \dots, p_k are decoded by the receiver. The smoothness exponent $\gamma_k^{(s)}$ is defined as the asymptotic decay rate of D_k , which is given by*

$$\gamma_k^{(s)} = - \lim_{n \rightarrow \infty} \frac{\log \Pr(D_k > n)}{n} \quad (2)$$

The relation (2) can also be stated as $\Pr(D > n) \doteq e^{-n\gamma}$ where \doteq stands for asymptotic equality defined in [23, Page 63]. For simplicity of analysis we define another delay exponent, the inter-delivery defined as follows. Theorem 1 shows the equivalence of the smoothness and the inter-delivery exponent for time-invariant schemes.

Definition 6 (Inter-delivery Exponent). *Let T_1 be the first inter-delivery time, that is the first time instant when one or more packets are decoded in-order. The inter-delivery exponent λ is defined as the asymptotic decay rate of T_1 , which is given by*

$$\lambda = - \lim_{n \rightarrow \infty} \frac{\log \Pr(T_1 > n)}{n} \quad (3)$$

In this paper we focus on time-invariant transmission schemes where the coding strategy is fixed across blocks of transmission, formally defined as follows.

Definition 7 (Time-invariant schemes). *A time-invariant scheme is represented by a vector $\mathbf{x} = [x_1, \dots, x_d]$ where x_i , for $1 \leq i \leq d$, are non-negative integers such that $\sum_i x_i = d$. In each block we transmit x_i independent linear combinations of the i lowest-index unseen packets in the stream, for $1 \leq i \leq d$.*

The above class of schemes is referred to as time-invariant because the vector \mathbf{x} is fixed across all blocks. Note that there is also no loss of generality in restricting the length of the vector \mathbf{x} to d . This is because each block can provide only up to d innovative coded packets, and hence there is no advantage in adding more than d unseen packets to the stream in a given block.

Theorem 1. *For a time-invariant scheme, the smoothness exponent $\gamma_k^{(s)}$ of packet s_k for any $k \leq \infty$ is equal to λ , the inter-delivery exponent.*

Thus, in the rest of the paper we study the trade-off between throughput τ and the inter-delivery exponent λ .

III. POINT-TO-POINT STREAMING

We first consider the extreme cases of immediate feedback and no feedback in Section III-A and Section III-B respectively. In Section III-C we propose coding schemes for the general case of block-wise feedback after every d slots.

A. Immediate Feedback

In the immediate feedback ($d = 1$) case, the source has complete knowledge of past erasures before transmitting each packet. We can show that a simple automatic-repeat-request (ARQ) scheme is optimal in both τ and λ . In this scheme, the source transmits the lowest index unseen packet, and repeats it until the packet successfully goes through the channel.

Since a new packet is received in every successful slot, the throughput $\tau = p$, the success probability of the erasure channel. The ARQ scheme is throughput-optimal because the throughput $\tau = p$ is equal to the information-theoretic capacity of the erasure channel [23]. Moreover, it also gives the optimal the inter-delivery exponent λ because one in-order packet is decoded in every successful slot. To find λ , first observe that the tail distribution of the time T_1 , the first inter-delivery time is,

$$\Pr(T_1 > n) = (1 - p)^n \quad (4)$$

Substituting this in Definition 6 we get the exponent $\lambda = -\log(1 - p)$. Thus, the trade-off for the immediate feedback case is $(\tau, \lambda) = (p, -\log(1 - p))$.

From this analysis of the immediate feedback case we can find limits on the range of achievable (τ, λ) for any feedback delay d . Since a scheme with immediate feedback can always simulate one with delayed feedback, the throughput and delay metrics (τ, λ) achievable for any feedback delay d must lie in the region $0 \leq \tau \leq p$, and $0 \leq \lambda \leq -\log(1 - p)$.

B. No Feedback

Now we consider the other extreme case ($d = \infty$), corresponding to when there is no feedback to the source. We propose a coding scheme that gives the best (τ, λ) trade-off among the class of full-rank codes, defined as follows.

Definition 8 (Full-rank Codes). *In slot n we transmit a linear combination of all packets s_1 to $s_{V[n]}$. We refer to $V[n]$ as the transmit index in slot n .*

Conjecture 1. *Given transmit index $V[n]$, there is no loss of generality in including all packets s_1 to $s_{V[n]}$.*

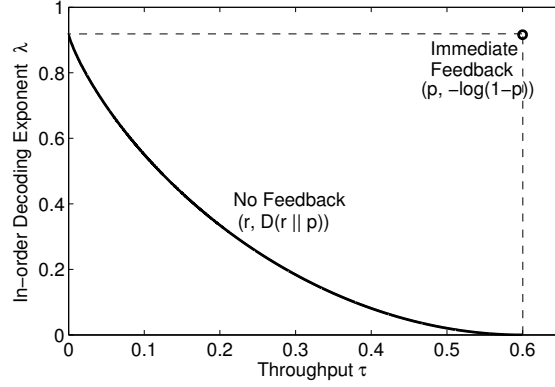


Fig. 1: The trade-off between inter-delivery exponent λ and throughput τ with success probability $p = 0.6$ for the immediate feedback ($d = 1$) and no feedback ($d = \infty$) cases.

We believe this conjecture is true because the packets are required in-order at the receiver. Thus, every packet s_j , $j < V[n]$ is required before packet $s_{V[n]}$ and there is no advantage in excluding s_j from the combination. Hence we believe that there is no loss of generality in restricting our attention to full-rank codes. A direct approach to verifying this conjecture would involve checking all possible channel erasure patterns.

Theorem 2. *The optimal throughput-smoothness trade-off among full-rank codes is $(\tau, \lambda) = (r, D(r||p))$ for all $0 \leq r < p$. It is achieved by the coding scheme with $V[n] = \lceil rn \rceil$ for all n .*

The term $D(r||p)$ is the binary information divergence function, which is defined for $0 < p, r < 1$ as

$$D(r||p) = r \log \frac{r}{p} + (1 - r) \log \frac{1 - r}{1 - p}, \quad (5)$$

where $0 \log 0$ is assumed to be 0. As $r \rightarrow 0$, $D(r||p)$ converges to $-\log(1 - p)$, which is the best possible λ as given in Section III-A.

Fig. 1 shows the (τ, λ) trade-off for the immediate feedback and no feedback cases, with success probability $p = 0.6$. The optimal trade-off with any feedback delay d lies in between these two extreme cases.

C. General Block-wise Feedback

In Section III-A and Section III-B we considered the extreme cases of immediate feedback ($d = 1$) and no feedback ($d = \infty$) respectively. We now analyze the (τ, λ) trade-off with general block-wise feedback delay of d slots. We restrict our attention to a class of coding schemes called time-invariant schemes, which are defined as follows.

Given a vector \mathbf{x} , define p_d , as the probability of decoding the first unseen packet during the block, and S_d as the number of innovative coded packets that are received during that block. We can express $\tau_{\mathbf{x}}$ and $\lambda_{\mathbf{x}}$ in terms of p_d and S_d as,

$$(\tau_{\mathbf{x}}, \lambda_{\mathbf{x}}) = \left(\frac{\mathbb{E}[S_d]}{d}, -\frac{1}{d} \log(1 - p_d) \right), \quad (6)$$

where we get throughput $\tau_{\mathbf{x}}$ by normalizing the $\mathbb{E}[S_d]$ by the number of slots in the slots. We can show that the probability $\Pr(T_1 > kd)$ of no in-order packet being decoded in k blocks is equal $(1 - p_d)^k$. Substituting this in (3) we get $\lambda_{\mathbf{x}}$.

Example 1. Consider the time-invariant scheme $\mathbf{x} = [1, 0, 3, 0]$ where block size $d = 4$. That is, we transmit 1 combination of the first unseen packet, and 3 combinations of the first 3 unseen packets. Fig. 2 illustrates this scheme for one channel realization. The probability p_d and $\mathbb{E}[S_d]$ are,

$$p_d = p + (1 - p) \binom{3}{3} p^3 (1 - p)^0 = p + (1 - p)p^3, \quad (7)$$

$$\mathbb{E}[S_d] = \sum_{i=1}^3 i \cdot \binom{4}{i} p^i (1 - p)^{4-i} + 3p^4 = 4p - p^4, \quad (8)$$

where in (8), we get i innovative packets if there are i successful slots for $1 \leq i \leq 3$. But if all 4 slots are successful we get only 3 innovative packets. We can substitute (7) and (8) in (6) to get the (τ, λ) trade-off.

Remark 1. Time-invariant schemes with different \mathbf{x} can be equivalent in terms of the (τ, λ) . In particular, given $x_1 \geq 1$, if any $x_i = 0$, and $x_{i+1} = w \geq 1$, then the scheme is equivalent to setting $x_i = 1$ and $x_{i+1} = w - 1$, keeping all other elements of \mathbf{x} the same. This is because the number of independent linear combinations in the block, and the probability of decoding

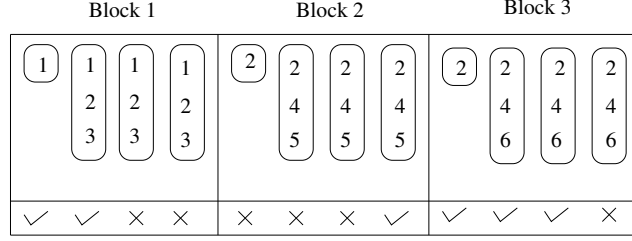


Fig. 2: Illustration of the time-invariant scheme $\mathbf{x} = [1, 0, 3, 0]$ with block size $d = 4$. Each bubble represents a coded combination, and the numbers inside it are the indices of the source packets included in that combination. The check and cross marks denote successful and erased slots respectively. The packets that are “seen” in each block are not included in the coded packets in future blocks.

the first unseen is preserved by this transformation. For example, $\mathbf{x} = [1, 1, 2, 0]$ gives the same (τ, λ) as $\mathbf{x} = [1, 0, 3, 0]$.

In Section III-A we saw that with immediate feedback, we can achieve $(\tau, \lambda) = (p, -\log(1 - p))$. However, with block-wise feedback we can achieve optimal τ (or λ) only at the cost of sacrificing the optimality of the other metric. We now find the best achievable τ (or λ) with optimal λ (or τ).

Claim 1 (Cost of Optimal Exponent λ). *With block-wise feedback after every d slots, and inter-delivery exponent $\lambda = -\log(1 - p)$, the best achievable throughput $\tau = (1 - (1 - p)^d)/d$.*

Proof: If we want to achieve $\lambda = -\log(1 - p)$, we require p_d in (6) to be equal to $1 - (1 - p)^d$. The only scheme that can achieve this is $\mathbf{x} = [d, 0, \dots, 0]$, where we transmit d copies of the first unseen packet. The number of innovative packets S_d received in every block is 1 with probability $1 - (1 - p)^d$, and zero otherwise. Hence, the best achievable throughput is $\tau = (1 - (1 - p)^d)/d$ with optimal $\lambda = -\log(1 - p)$. ■

This result gives us insight on how much bandwidth (which is proportional to $1/\tau$) is needed for a highly delay-sensitive application that needs λ to be as large as possible.

Claim 2 (Cost of Optimal Throughput τ). *With block-wise feedback after every d slots, and*

throughput $\tau = p$, the best achievable inter-delivery exponent is $\lambda = -\log(1 - p)/d$.

Proof: If we want to achieve $\tau = p$, we need to guarantee an innovation packet in every successful slot. The only time invariant scheme that ensures this is $\mathbf{x} = [1, 0, \dots, 0, d - 1]$, or its equivalent vectors \mathbf{x} as given by Remark 1. With $\mathbf{x} = [1, 0, \dots, 0, d - 1]$, the probability of decoding the first unseen packet is $p_d = p$. Substituting this in (6) we get $\lambda = -\log(1 - p)/d$, the best achievable λ when $\tau = p$. ■

Tying back to Fig. 1, Claim 1 and Claim 2 correspond to moving leftwards and downwards along the dashed lines from the optimal trade-off $(p, -\log(1 - p))$. From Claim 1 and Claim 2 we see that both τ and λ are $\Theta(1/d)$, keeping the other metric optimal.

For any given throughput τ , our aim is to find the coding scheme that maximizes λ . We first prove that any convex combination of achievable points (τ, λ) can be achieved.

Lemma 1 (Combining of Time-invariant Schemes). *By randomizing between time-invariant schemes $\mathbf{x}^{(i)}$ for $1 \leq i \leq B$, we can achieve the throughput-smoothness trade-off given by any convex combination of the points $(\tau_{\mathbf{x}^{(i)}}, \lambda_{\mathbf{x}^{(i)}})$.*

The proof of Lemma 1 is deferred to the Appendix. The main implication of Lemma 1 is that, to find the best (τ, λ) trade-off, we only have to find the points $(\tau_{\mathbf{x}}, \lambda_{\mathbf{x}})$ that lie on the convex envelope of the achievable region spanned by all possible \mathbf{x} .

For general d , it is hard to search for the $(\tau_{\mathbf{x}}, \lambda_{\mathbf{x}})$ that lie on the optimal trade-off. We propose a set of time-invariant schemes that are easy to analyze and give a good (τ, λ) trade-off. In Theorem 3 we give the (τ, λ) trade-off for the proposed codes and show that for $d = 2$ and $d = 3$, it is the best trade-off among all time-invariant schemes.

Definition 9 (Proposed Codes for general d). *For general d , we propose using the time-invariant schemes with $x_1 = a$ and $x_{d-a+1} = d - a$, for $a = 1, \dots, d$.*

In other words, in every block of d slots, we transmit the first unseen packet a times, followed by $d - a$ combinations of the first $d - a + 1$ unseen packets. These schemes span the (τ, λ) trade-off as a varies from 1 to d , with a higher value of a corresponding to higher λ and lower τ . In particular, observe that the $a = d$ and $a = 1$ codes correspond to codes given in the proofs

of Claim 1 and Claim 2.

Theorem 3 (Throughput-Smoothness Trade-off for General d). *The codes proposed in Definition 9 give the trade-off points*

$$(\tau, \lambda) = \left(\frac{1 - (1 - p)^a + (d - a)p}{d}, -\frac{a}{d} \log(1 - p) \right). \quad (9)$$

for $a = 1, \dots, d$.

Proof: To find the (τ, λ) trade-off points, we first evaluate $\mathbb{E}[S_d]$ and p_d . With probability $1 - (1 - p)^a$ we get 1 innovative packet from the first a slots in a block. The number of innovative packets received in the remaining $d - a$ slots is equal to the number of successful slots. Thus, the expected number of innovative coded packets received in the block is

$$\mathbb{E}[S_d] = 1 - (1 - p)^a + (d - a)p \quad (10)$$

If the first a slots in the block are erased, the first unseen packet cannot be decoded, even if all the other slots are successful. Hence, we have $p_d = 1 - (1 - p)^a$. Substituting $\mathbb{E}[S_d]$ and p_d in (6), we get the trade-off in (9). \blacksquare

By Lemma 1, we can achieve any convex combination of the (τ, λ) points in (9). In Lemma 2 we show that for $d = 2$ and $d = 3$ this is the best trade-off among all time-invariant schemes.

Lemma 2. *For $d = 2$ and $d = 3$, the codes proposed in Definition 9 give the best (τ, λ) trade-off among all time-invariant schemes.*

Fig. 3 shows the trade-off given by (9) for different values of d . We observe that the trade-off becomes significantly worse as d increases. Thus we can imply that frequent feedback to the source is important in delay-sensitive applications to ensure fast in-order delivery of packets. As $d \rightarrow \infty$, and $a = \alpha d$, the trade-off converges to $((1 - \alpha)p, -\alpha \log(1 - p))$ for $0 \leq \alpha \leq 1$, which is the line joining $(0, -\log(1 - p))$ and $(p, 0)$. It does not converge to the $(r, D(r||p))$ curve without feedback because we consider that d goes to infinity slower than the n used to evaluate the asymptotic exponent of $\Pr(T_1 > n)$.

By Lemma 2 the proposed codes give the best trade-off among all time-invariant schemes. Numerical results suggest that even for general d these schemes give a trade-off that is close to the best trade-off among all time-invariant schemes.

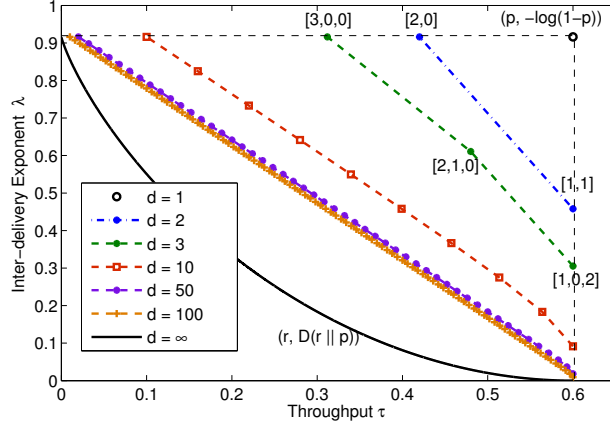


Fig. 3: The throughput-smoothness trade-off of the suggested coding schemes in Definition 9 for $p = 0.6$ and various values of block-wise feedback delay d . The trade-off becomes significantly worse as d increases. The point labels on the $d = 2$ and $d = 3$ trade-offs are \mathbf{x} vectors of the corresponding codes.

Thus, in this section we analyzed how block-wise feedback affects the trade-off between throughput τ and inter-delivery exponent λ , which measures the burstiness in-order delivery in streaming communication. Our analysis gives us the insight that frequent feedback is crucial for fast in-order packet delivery. Given that feedback comes in blocks of d slots, we present a spectrum of coding schemes that span different points on the (τ, λ) trade-off. Depending upon the delay-sensitivity and bandwidth limitations of the applications, these codes provide the flexibility to choose a suitable operating point on trade-off.

IV. MULTICAST STREAMING

In this section we move to the multicast streaming scenario. Since each user decodes a different set of packets, the in-order packet required by one user may be redundant to the other user, causing the latter to lose throughput if that packet is transmitted. In Section IV-A we identify the structure of coding schemes required to simultaneously satisfy the requirements of multiple users. In Section IV-B we use this structure to find the best coding scheme for the two user case, where one user is always given higher priority. In Section IV-C we generalize this scheme to allow tuning the level of priority given to each user. The analysis of both these cases is based

on a new Markov chain model of packet decoding.

For this section, we focus on the case where each user provides immediate feedback to the source ($d = 1$).

Remark 2. *For the no feedback case ($d = \infty$) we can extend Theorem 2 to show that the optimal throughput-smoothness trade-off for user U_k , $k = 1, 2, \dots, K$ among full-rank codes is $(\tau_k, \lambda_k) = (r, D(r||p_k))$, if $0 \leq r \leq p_k$. If $r > p_k$ then $\lambda_k = 0$ for user U_k . Since we are transmitting a common stream, the rate r of adding new packets is same for all users.*

The general d case is hard to analyze and open for future work.

A. Structure of Coding Schemes

The best possible trade-off is $(\tau_i, \lambda_i) = (p_i, -\log(1 - p_i))$, and it can be achieved when there is only one user, and the source uses a simple Automatic-repeat-request (ARQ) protocol where it keeps retransmitting the earliest undecoded packet until that packet is decoded. In this paper our objective is to design coding strategies to maximize τ and λ for the two user case. For two or more users we can show that it is impossible to achieve the optimal trade-off $(\tau, \lambda) = (p_i, -\log(1 - p_i))$ simultaneously for all users. We now present code structures that maximize throughput and inter-delivery exponent of the users.

Claim 3 (Include only Required Packets). *In a given slot, it is sufficient for the source to transmit a combination of packets s_{r_i} for $i \in \mathcal{I}$ where \mathcal{I} is some subset of $\{1, 2, \dots, K\}$.*

Proof: Consider a candidate packet s_c where $c \neq r_i$ for any $1 \leq i \leq K$. If $c < r_i$ for all i , then s_c has been decoded by all users, and it need not be included in the combination. For all other values of c , there exists a required packet s_{r_i} for some $i \in \{1, 2, \dots, K\}$ that, if included instead of s_c , will allow more users to decode their required packets. Hence, including that packet instead of s_c gives a higher exponent λ . ■

Claim 4 (Include only Decodable Packets). *If a coded combination already includes packets s_{r_i} with $i \in \mathcal{I}$, and U_j , $j \notin \mathcal{I}$ has not decoded all s_{r_i} for $i \in \mathcal{I}$, then a scheme that does not include s_{r_j} in the combination gives a better throughput-smoothness trade-off than a scheme that does.*

Proof: If U_j has not decoded all s_{r_i} for $i \in \mathcal{I}$, the combination is innovative but does not help decoding an in-order packet, irrespective of whether s_{r_j} is included in the combination. However, if we do not include packet s_{r_j} , U_j may be able to decode one of the packets s_{r_i} , $i \in \mathcal{I}$, which can save it from out-of-order packet decoding in a future slot. Hence excluding s_{r_j} gives a better throughput-smoothness trade-off. ■

Example 2. Suppose we have three users U_1 , U_2 , and U_3 . User U_1 has decoded packets s_1 , s_2 , s_3 and s_5 , user U_2 has decoded s_1 , s_3 , and s_4 , and user U_3 has decoded s_1 , s_2 , and s_5 . The required packets of the three users are s_4 , s_2 and s_3 respectively. By Claim 3, the optimal scheme should transmit a linear combination of one or more of these packets. Suppose we construct combination of s_4 and s_2 and want to decide whether to include s_3 or not. Since user U_3 has not decoded s_4 , we should not include s_3 as implied by Claim 4.

The choice of the initial packets in the combination is governed by a priority given to each user in that slot. Claims 3 and 4 imply the following code structure for the two user case.

Proposition 1 (Code Structure for the Two User Case). *Every achievable trade-off between throughput and inter-delivery exponent can be obtained by a coding scheme where the source transmits s_{r_1} , s_{r_2} or the exclusive-or, $s_{r_1} \oplus s_{r_2}$ in each slot. It transmits $s_{r_1} \oplus s_{r_2}$ if and only if $r_1 \neq r_2$, and U_1 has decoded s_{r_2} or U_2 has decoded s_{r_1} .*

In the rest of this section we analyze the two user case and focus on coding schemes as given by Proposition 1.

B. Optimal Performance for One of Two Users

In this section we consider that the source always gives priority to one user, called the primary user. We determine the best achievable throughput-smoothness trade-off for a secondary user that is “piggybacking” on such a primary user. For simplicity of notation, let $a \triangleq p_1 p_2$, $b \triangleq p_1(1-p_2)$, $c \triangleq (1-p_1)p_2$ and $d \triangleq (1-p_1)(1-p_2)$, the probabilities of the four possible erasure patterns.

Without loss of generality, suppose that U_1 is the primary user, and U_2 is the secondary user. Recall that ensuring optimal performance for U_1 implies achieving $(\tau_1, \lambda_1) = (p_1, -\log(1-p_1))$. While ensuring this, the best throughput-smoothness trade-off for user U_2 is achieved by the

Time	Sent	U_1	U_2
1	s_1	s_1	\times
2	s_2	\times	s_2
3	$s_1 \oplus s_2$	s_2	s_1
4	s_3	s_3	\times
5	s_4	s_4	s_4

Fig. 4: Illustration of the optimal coding scheme when the source always give priority to user U_1 . The third and fourth columns show the packets decoded at the two users. Cross marks indicate erased slots for the corresponding user.

coding scheme given by Claim 5 below.

Claim 5 (Optimal Coding Scheme). *A coding scheme where the source transmits $s_{r_1} \oplus s_{r_2}$ if U_2 has already decoded s_{r_1} , and otherwise transmits s_{r_1} , gives the best achievable (τ_2, λ_2) trade-off while ensuring optimal (τ_1, λ_1) .*

Proof: Since U_1 is the primary user, the source must include its required packet s_{r_1} in every coded combination. By Proposition 1, if the source transmits $s_{r_1} \oplus s_{r_2}$ if U_2 has already decoded s_{r_1} , and transmits s_{r_1} otherwise, we get the best achievable throughput-smoothness trade-off for U_2 . ■

Fig. 4 illustrates this scheme for one channel realization.

Packet decoding at the two users with the scheme given by Claim 5 can be modeled by the Markov chain shown in Fig. 5. The state index i can be expressed in terms of the number of gaps in decoding of the users, defined as follows.

Definition 10 (Number of Gaps in Decoding). *The number of gaps in U_i 's decoding is the number of undecoded packets of U_i with indices less than $r_{\max} = \max_i r_i$.*

In other words, the number of gaps is the amount by which a user U_i lags behind the user that is leading the in-order packet decoding. The state index i , for $i \geq -1$ is equal to the number of gaps in decoding at U_2 , minus that for U_1 . Since the source gives priority to U_1 , it always has zero gaps in decoding, except when there is a $c = p_2(1 - p_1)$ probability erasure in state 0,

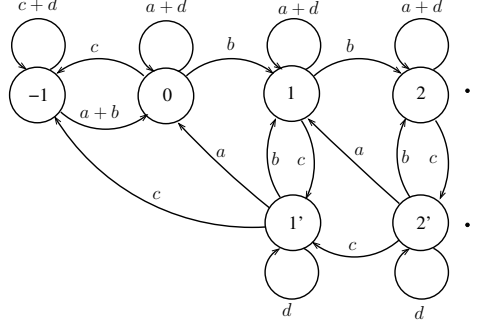


Fig. 5: Markov chain model of packet decoding with the coding scheme given by Claim 5, where U_1 is the primary user. The state index i represents the number of gaps in decoding of U_2 minus that for U_1 . The states i' are the advantage states where U_2 gets a chance to decode its required packet.

which causes the system goes to state -1 . The states i' for $i \geq 1$ are called “advantage” states and are defined as follows.

Definition 11 (Advantage State). *The system is in an advantage state when $r_1 \neq r_2$, and U_2 has decoded s_{r_1} but U_1 has not.*

By Claim 5, the source transmits $s_{r_1} \oplus s_{r_2}$ when the system is in an advantage state i' , and it transmits s_{r_1} when the system is in state i for $i \geq -1$. We now describe the state transitions of this Markov chain. First observe that with probability $d = (1 - p_1)(1 - p_2)$, both users experience erasures and the system transitions from any state to itself. When the system is in state -1 , the source transmits s_{r_1} . Since s_{r_1} has been already decoded by U_2 , the probability $c = p_2(1 - p_1)$ erasure also keeps the system in the same state. If the channel is successful for U_1 , which occurs with probability $p_1 = a + b$, it fills its decoding gap and the system goes to state 0.

The source transmits s_{r_1} in any state i , $i \geq 1$. With probability $a = p_1 p_2$, both users decode s_{r_1} , and hence the state index i remains the same. With probability $b = p_1(1 - p_2)$, U_1 receives s_{r_1} but U_2 does not, causing a transition to state $i + 1$. With probability $c = (1 - p_1)p_2$, U_2 receives s_{r_1} and U_1 experiences an erasure due to which the system moves to the advantage state i' . When the system is an advantage state, having decoded s_{r_1} gives U_2 an advantage because it can use $s_{r_1} \oplus s_{r_2}$ transmitted in the next slot to decode s_{r_2} . From state i' , with probability a ,

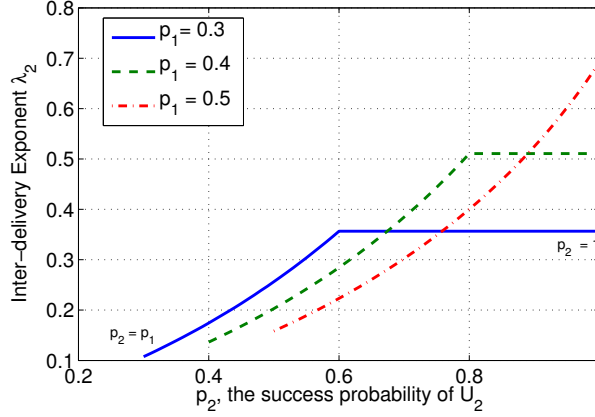


Fig. 6: Plots of the inter-delivery exponent λ_2 of the piggybacking user U_2 , versus the success probability p_2 throughput τ_2 . The value of p_2 varies from p_1 to 1 on each curve. The exponent saturates at $-\log(1 - p_1)$, which is equal to λ_1 , the exponent of the primary user U_1 .

U_1 decodes s_{r_1} and U_2 decodes s_{r_2} , and the state transitions to $i - 1$. With probability c , U_2 decodes s_{r_2} , but U_1 does not decode s_{r_1} . Thus, the system goes to state $(i - 1)'$, except when $i = 1$, where it goes to state 0.

Claim 6. *The Markov chain in Fig. 5 is positive-recurrent and has unique steady-state distribution if and only if $b < c$, which is equivalent to $p_1 < p_2$.*

Lemma 3 (Trade-off for the Piggybacking user). *When the source always gives priority to user U_1 it achieves the optimal trade-off $(\tau_1, \lambda_1) = (p_1, -\log(1 - p_1))$. The scheme in Claim 5 gives the best achievable (τ_2, λ_2) trade-off for piggybacking user U_2 . The throughput τ_2 is given by*

$$\tau_2 = p_1 \quad \text{if } p_2 > p_1. \quad (11)$$

If $p_2 \leq p_1$, τ_2 cannot be evaluated using our Markov chain analysis. The inter-delivery exponent of U_2 for any p_1 and p_2 is given by

$$\lambda_2 = -\log \left(\max \left(\frac{\left(1 - c + d + \sqrt{(1 - c + d)^2 + 4(bc + cd - d)} \right)}{2}, 1 - p_1 \right) \right). \quad (12)$$

In Fig. 6 we plot the inter-delivery exponent λ_2 versus p_2 , which increases from p_1 to 1 along each curve. The inter-delivery exponent λ_2 increases with p_2 , but saturates at $-\log(1 - p_1)$, the

inter-delivery exponent λ_1 of the primary user. Since U_2 is the secondary user it cannot achieve faster in-order delivery than the primary user U_1 .

C. General Throughput-Smoothness Trade-offs

For the general case, we propose coding schemes that can be combined to tune the priority given to each of the two users and achieve different points on their throughput-smoothness trade-offs.

Let $r_{\max} = \max(r_1, r_2)$ and $r_{\min} = \min(r_1, r_2)$, where r_1 and r_2 are the indices of the required packets of the two users. We refer to the user with the higher index r_i as the leader(s) and the other user as the lagger. Thus, U_1 is the leader and U_2 is the lagger when $r_1 > r_2$. If $r_1 = r_2$, without loss of generality we consider U_1 as the leader.

Definition 12 (Priority- (q_1, q_2) Codes). *If the lagger U_i has not decoded packet $s_{r_{\max}}$, the source transmits $s_{r_{\min}}$ with probability q_i and $s_{r_{\max}}$ otherwise. If the lagger has decoded $s_{r_{\max}}$, the source transmits $s_{r_{\max}} \oplus s_{r_{\min}}$.*

Note that the code given in Claim 5, where the source always gives priority to user U_1 is a special case of priority- (q_1, q_2) codes with $(q_1, q_2) = (1, 0)$. Another special case is $(q_1, q_2) = (0, 0)$ which is a greedy coding scheme that always favors the user which is ahead in in-order delivery. The greedy coding scheme ensures throughput optimality to both users, i.e. $\tau_1 = p_1$ and $\tau_2 = p_2$.

Remark 3. *A generalization of priority- (q_1, q_2) codes is to consider priorities $q_1^{(i)}$ and $q_2^{(i)}$ that depend on the state i of the Markov chain. A special case of this is $q_1^{(i)} = 1$ for all states $i \geq -M$, and $q_2^{(i)} = 1$ for all states $j \leq N$ for integers $M, N > 0$. This scheme corresponds to putting hard boundaries on both sides of the Markov chain, and was analyzed in [22].*

The Markov model of packet decoding with a priority- (q_1, q_2) code is as shown in Fig. 7, which is a two-sided version of the Markov chain in Fig. 5. Same as in Fig. 5, the index i of a state i of the Markov chain is the number of gaps in decoding of U_2 minus that for U_1 . User U_1 is the leader when the system is in state $i \geq 1$ and U_2 is the leader when $i \leq -1$, and both users are leaders when $i = 0$. The system is in the advantage state i' if packet is decoded by

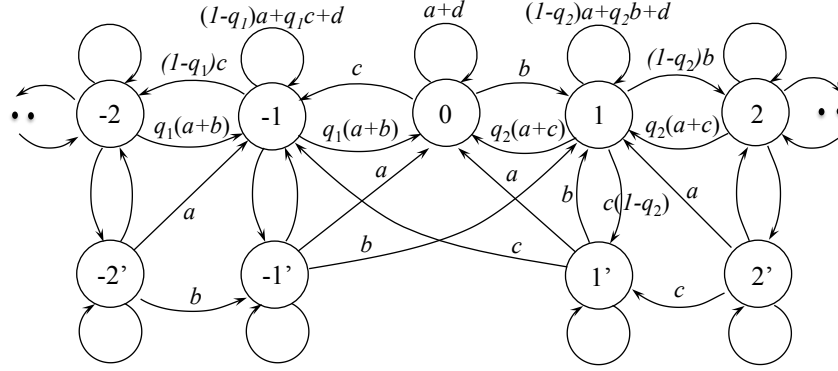


Fig. 7: Markov chain model of packet decoding with the priority- (q_1, q_2) coding scheme given by Definition 12. The state index i represents the number of gaps in decoding if U_2 compared to U_1 and q_i is the probability of giving priority to the U_i when it is the lagger, by transmitting its required packet s_{r_i} .

the lagger but not the leader.

For simplicity of representation we define the notation $\bar{d} \triangleq 1 - d$, $\bar{q}_1 \triangleq 1 - q_1$ and $\bar{q}_2 \triangleq 1 - q_2$.

Lemma 4 ((τ, λ) Trade-offs with Priority- (q_1, q_2) codes). *Let $\mu = \pi_{i-1}/\pi_i$ for $i \leq -1$. Then the priority- (q_1, q_2) codes given by Definition 12 give the following throughput for U_2 .*

$$\tau_2 = p_2 \left(1 - \frac{q_1 \pi_{-1}}{1 - \mu} \right) \quad \text{if } \mu < 1 \quad (13)$$

If $\mu > 1$ then τ_2 cannot be evaluated using our Markov chain analysis. On the other hand, the inter-delivery exponent can be evaluated for any μ as given by

$$\lambda_2 = -\log \max \left(d + q_1 c + \bar{q}_1 b, \frac{\left(2d + \bar{q}_2 a + b + \sqrt{(2d + \bar{q}_2 a + b)^2 - 4(d(b + d) + \bar{q}_2(da - bc))} \right)}{2} \right) \quad (14)$$

Similarly, let $\rho = \pi_{i+1}/\pi_i$ for $i \geq 1$. If $\rho < 1$, the expressions for throughput τ_1 and inter-delivery exponent λ_1 of U_1 are same as (13) and (14) with b and c , and q_1 and q_2 interchanged, π_{-1} replaced by π_1 , and μ replaced by ρ .

Fig. 8 shows the throughput-smoothness trade-offs of the two users as q_2 varies, when $q_1 = 1$, $p_1 = 0.5$ and $p_2 = 0.4$. To stabilize the right-side of the Markov chain in Fig. 7 for these

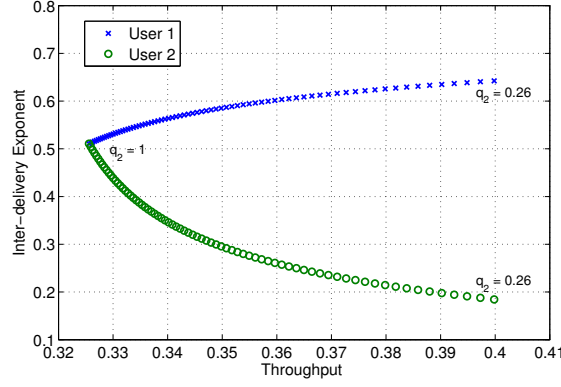


Fig. 8: Plot of the throughput-smoothness trade-off for $q_1 = 1$ and as q_2 varies. The success probabilities $p_1 = 0.5$ and $p_2 = 0.4$.

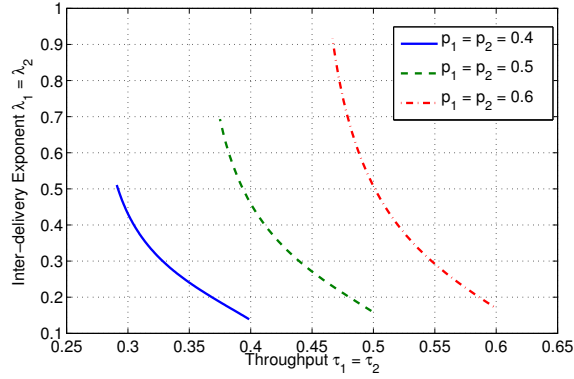


Fig. 9: Plot of the throughput-smoothness trade-off for different values $p_1 = p_2$. On each curve, $q_1 = q_2$ varies from 0 from 1.

parameters we require q_2 to be at least 0.25. As q_2 increases from 0.26 to 1 in Fig. 8 we observe that U_2 gains in smoothness, at the cost of the smoothness of U_1 . Also, both users lose throughput when q_2 increases.

In Fig. 9 we show the effect of increasing q_1 and q_2 simultaneously for different values of $p_1 = p_2$. As $q_1 = q_2$ increases, we get a better inter-delivery exponent for both users, but at the cost of loss of throughput.

V. CONCLUDING REMARKS

A. Major Implications

In this paper we consider the problem of streaming over an erasure channel when the packets are required in-order by the receiver application. We investigate the trade-off between the throughput and the smoothness of in-order packet delivery. The smoothness is measured in terms of the smoothness exponent, which we show to be equivalent to an easy-to-analyze metric called the inter-delivery exponent.

We first study the effect of block-wise feedback affects the throughput and smoothness of packet delivery in point-to-point streaming. Our analysis shows that frequent feedback drastically improves the smoothness, for the same throughput. We present a spectrum of coding schemes that span different points on the throughput-smoothness trade-off. Depending upon the delay-sensitivity and bandwidth limitations of the applications, one can choose a suitable operating point on this trade-off.

Next we consider the problem where multiple users are streaming data from the source over a broadcast channel, over independent erasure channels with immediate feedback. Since different users decode different sets of packets, the source has to strike balance between giving priority to ensuring in-order packet decoding at each of the users. We study the inter-dependence between the throughput-delay trade-offs for the case of two users and develop coding schemes to tune the priority given to each user.

B. Future Perspectives

In this paper, we assume strict in-order packet delivery, without allowing packet dropping, which can improve the smoothness exponent. Determining how significant this improvement is remains to be explored. Another possible future research direction is to extend the multicast streaming analysis to more users using the Markov chain-based techniques developed here for the two user case. A broader research direction is to consider the problem of streaming from distributed sources that are storing overlapping sets of the data. Using this diversity can improve the delay in decoding each individual packet.

APPENDIX

Proof of Theorem 1: The in-order decoding delay D_k of packet s_k can be expressed as a sum of inter-delivery times as follows.

$$D_k = T_1 + T_2 + \dots T_W \quad (15)$$

where W is the number of in-order delivery instants until packets s_1, \dots, s_k are decoded. The random variable W can take values $1 \leq W \leq k$, since multiple packets may be decoded at one in-order delivery instants. Note that successive inter-delivery times T_1, T_2, \dots, T_W are not i.i.d. The tail probability $\Pr(T_1 > t)$ of the first inter-delivery time is,

$$\Pr(T_1 > t) \geq \Pr(T_i > t) \text{ for all integers } i, t \geq 0. \quad (16)$$

This is because during the first inter-delivery time T_1 we start with no prior information. During time T_1 , the receiver may collect coded combinations that it is not able to decode. For a time-invariant scheme, these coded combinations can result in faster in-order decoding and hence a smaller T_i for $i > 1$.

We now find lower and upper bounds on $\Pr(D_k \geq n)$ to find the decay rate of D_k . The lower bound can be derived as follows.

$$\Pr(D_k \geq n) = \mathbb{E}_W [\Pr(T_1 + T_2 + \dots T_W \geq n)] \quad (17)$$

$$\geq \Pr(T_1 \geq n) \quad (18)$$

$$\doteq e^{-\lambda n} \quad (19)$$

where (19) follows from Definition 6. Now we derive an upper bound on $\Pr(D_k > n)$.

$$\Pr(D_k \geq n) = \mathbb{E}_W [\Pr(T_1 + T_2 + \dots + T_W \geq n)] \quad (20)$$

$$\leq \mathbb{E}_W \left[\Pr(T_1^{(1)} + T_1^{(2)} + \dots + T_1^{(W)} \geq n) \right] \quad (21)$$

$$= \mathbb{E}_W \left[\sum_{n_i: \sum_{i=1}^W n_i = n} \prod_{i=1}^W \Pr(T_1^{(i)} > n_i) \right] \quad (22)$$

$$\doteq \mathbb{E}_W \left[\sum_{n_i: \sum_{i=1}^W n_i = n} e^{-\lambda(n_1 + \dots + n_W)} \right] \quad (23)$$

$$= \mathbb{E}_W \left[\binom{n+W-1}{W-1} e^{-\lambda n} \right] \quad (24)$$

$$\doteq \mathbb{E}_W [e^{-\lambda n}] \quad (25)$$

$$= e^{-\lambda n} \quad (26)$$

where $T_1^{(i)}$ are i.i.d. samples from the probability distribution of the first inter-delivery time T_1 . By (16), replacing T_i by $T_1^{(i)}$ gives an upper bound on the probability $\Pr(D_k \geq n)$. Since $T_1^{(i)}$ are i.i.d. we can express the upper bound as a product of tail probabilities in (22), where n_i are non-negative integers summing to n . By (3), each term in the product in (22) asymptotically decays at rate λ . Thus we get (23) and (24). Since $W \leq k \ll n$, the binomial coefficient decays subexponentially, and we get (26).

From (19) and (26) we can conclude that the asymptotic decay rate $\lambda_k^{(s)}$ for any k is equal to the inter-delivery exponent λ . ■

Proof of Theorem 2: We first show that the scheme with transmit index $V[n] = \lceil rn \rceil$ in time slot n achieves the trade-off $(\tau, \lambda) = (r, D(r||p))$. Then we prove the converse by showing that no other full-rank scheme gives a better trade-off.

Achievability Proof: Consider the scheme with transmit index $V[n] = \lceil rn \rceil$, where r represents the rate of adding new packets to the transmitted stream. The rate of adding packets is below the capacity of the erasure channel. Thus it is easy to see that the throughput $\tau = r$. Let $E[n]$ be the number of combinations, or equations received until time n . It follows the binomial distribution with parameter p . All packets $s_1 \dots s_{V[n]}$ are decoded when $E[n] \geq V[n]$. Define

the event $G_n = \{E[j] < V[j] \text{ for all } 1 \leq j \leq n\}$, that there is no packet decoding until slot n . The tail distribution of the first inter-delivery time T_1 is,

$$\begin{aligned} \Pr(T_1 > n) &= \sum_{k=0}^{\lceil nr \rceil - 1} \Pr(E[n] = k) \Pr(G_n | E[n] = k), \\ &= \sum_{k=0}^{\lceil nr \rceil - 1} \binom{n}{k} p^k (1-p)^{n-k} \Pr(G_n | E[n] = k), \end{aligned}$$

where $\Pr(G_n | E[n] = k) = 1 - k/n$ as given by the Generalized Ballot theorem in [24, Chapter 4].

Hence it is sub-exponential and does not affect the exponent of $\Pr(T_1 > n)$ and we have

$$\Pr(T_1 > n) \doteq \sum_{k=0}^{\lceil nr \rceil - 1} \binom{n}{k} p^k (1-p)^{n-k}, \quad (27)$$

$$\doteq \binom{n}{\lceil nr \rceil - 1} p^{\lceil nr \rceil - 1} (1-p)^{n - \lceil nr \rceil + 1}, \quad (28)$$

$$\doteq e^{-nD(r||p)}, \quad (29)$$

where in (27) we take the asymptotic equality \doteq to find the exponent of $\Pr(T_1 > n)$, and remove the $\Pr(G_n | E[n] = k)$ term because it is sub-exponential. In (28), we only retain the $k = \lceil nr \rceil - 1$ term from the summation because for $r \leq p$, that term asymptotically dominates other terms. Finally, we use the Stirlings approximation $\binom{n}{k} \approx e^{nH(k/n)}$ to obtain (29).

Converse Proof: First we show that the transmit index $V[n]$ of the optimal full-rank scheme should be non-decreasing in n . Given any scheme, we can permute the order of transmitting the coded packets such that $V[n]$ is non-decreasing in n . This does not affect the throughput τ , but it can improve the inter-delivery exponent λ because decoding can occur sooner when the initial coded packets include fewer source packets.

We now show that it is optimal to have $V[n] = \lceil rn \rceil$, where we add new packets to the transmitted stream at a constant rate r . Suppose a full-rank scheme uses rate r_i for n_i slots for

all $1 \leq i \leq L$, such that $\sum_{i=0}^L n_i = n$ and $\sum_{i=1}^L n_i r_i = nr$. Then, the tail distribution of T_1 is,

$$\Pr(T_1 > n) = \sum_{k=0}^{\lceil \sum_{i=1}^L n_i r_i \rceil - 1} \Pr(E[n] = k) \Pr(G_n | E[n] = k), \quad (30)$$

$$\doteq \sum_{k=0}^{\lceil nr \rceil - 1} \binom{n}{k} p^k (1-p)^{n-k}, \quad (31)$$

$$\doteq e^{-nD(r||p)}. \quad (32)$$

Varying the rate of adding packets affects the term $\Pr(G_n | E[n] = k)$ in (30), but it is still $\omega(1/n)$ and we can eliminate it when we take the asymptotic equality in (31). As a result, the in-order delay exponent is same as that if we had a constant rate r of adding new packets to the transmitted stream. Hence we have proved that no other full-rank scheme can achieve a better (τ, λ) trade-off than $V[n] = \lceil nr \rceil$ for all n . ■

Proof of Lemma 1: Here we prove the result for $B = 2$, that is randomizing between two schemes. It can be extended to general B using induction. Given two time-invariant schemes $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ that achieve the throughput-delay trade-offs $(\tau_{\mathbf{x}^{(1)}}, \lambda_{\mathbf{x}^{(1)}})$ and $(\tau_{\mathbf{x}^{(2)}}, \lambda_{\mathbf{x}^{(2)}})$ respectively, consider a randomized strategy where, in each block we use the scheme $\mathbf{x}^{(1)}$ with probability μ and scheme $\mathbf{x}^{(2)}$ otherwise. Then, it is easy to see that the throughput on the new scheme is $\tau = \mu\tau_{\mathbf{x}^{(1)}} + (1 - \mu)\tau_{\mathbf{x}^{(2)}}$.

Now we prove the inter-delivery exponent λ is also a convex combinations of $\lambda_{\mathbf{x}^{(1)}}$ and $\lambda_{\mathbf{x}^{(2)}}$. Let p_{d_1} and p_{d_2} be the probabilities of decoding the first unseen packet in a block using scheme $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ respectively. Suppose in an interval with k blocks, we use scheme $\mathbf{x}^{(1)}$ for h blocks, and scheme $\mathbf{x}^{(2)}$ in the remaining blocks, we have

$$\Pr(T_1 > kd) = (1 - p_{d_1})^h (1 - p_{d_2})^{k-h}. \quad (33)$$

Using this we can evaluate λ as,

$$\lambda = \lambda_{\mathbf{x}^{(1)}} \lim_{k \rightarrow \infty} \frac{h}{k} + \lambda_{\mathbf{x}^{(2)}} \lim_{k \rightarrow \infty} \frac{k-h}{k} \quad (34)$$

$$= \mu\lambda_{\mathbf{x}^{(1)}} + (1 - \mu)\lambda_{\mathbf{x}^{(2)}} \quad (35)$$

where we get (34) using (6). As $k \rightarrow \infty$, by the weak law of large numbers, the fraction h/k converges to μ . ■

Proof of Lemma 2: When $d = 2$ there are only two possible time-invariant schemes $\mathbf{x} = [2, 0]$ and $[1, 1]$ that give unique (τ, λ) . By Remark 1, all other \mathbf{x} are equivalent to one of these vectors in terms (τ, λ) . The vectors $\mathbf{x} = [2, 0]$ and $[1, 1]$ correspond to the $a = 1$ and $a = 2$ codes proposed in Definition 9. Hence, the line joining their corresponding (τ, λ) points, as shown in Fig. 3, is the best trade-off for $d = 2$.

When $d = 3$ there are four time-invariant schemes $\mathbf{x}^{(1)} = [1, 0, 2]$, $\mathbf{x}^{(2)} = [2, 1, 0]$, $\mathbf{x}^{(3)} = [1, 2, 0]$ and $\mathbf{x}^{(4)} = [3, 0, 0]$ that give unique (τ, λ) , according to Definition 7 and Remark 1. The vectors $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$ and $\mathbf{x}^{(4)}$ correspond to the codes with $a = 1, 2, 3$ in Definition 9. The throughput-delay trade-offs $(\tau_{\mathbf{x}^{(i)}}, \lambda_{\mathbf{x}^{(i)}})$ for $i = 1, 2, 4$ achieved by these schemes are given by (9). From Claim 1 and Claim 2 we know that $(\tau_{\mathbf{x}^{(1)}}, \lambda_{\mathbf{x}^{(1)}})$ and $(\tau_{\mathbf{x}^{(4)}}, \lambda_{\mathbf{x}^{(4)}})$ have to be on the optimal trade-off. By comparing the slopes of the lines joining these points we can show that the point $(\tau_{\mathbf{x}^{(2)}}, \lambda_{\mathbf{x}^{(2)}})$ lies above the line joining $(\tau_{\mathbf{x}^{(1)}}, \lambda_{\mathbf{x}^{(1)}})$ and $(\tau_{\mathbf{x}^{(4)}}, \lambda_{\mathbf{x}^{(4)}})$ for all p . Fig. 3 illustrates this for $p = 0.6$. For the scheme with $\mathbf{x}^{(3)} = [1, 2, 0]$, we have

$$(\tau_{\mathbf{x}^{(3)}}, \lambda_{\mathbf{x}^{(3)}}) = ((3p - p^3)/3, -(\log(1 - p)^2(1 + p))/3).$$

Again, by comparing the slopes of the lines joining $(\tau_{\mathbf{x}^{(i)}}, \lambda_{\mathbf{x}^{(i)}})$ for $i = 1, \dots, 4$ we can show that for all p , $(\tau_{\mathbf{x}^{(3)}}, \lambda_{\mathbf{x}^{(3)}})$ lies below the piecewise linear curve joining $(\tau_{\mathbf{x}^{(i)}}, \lambda_{\mathbf{x}^{(i)}})$ for $i = 1, 2, 4$. ■

Proof on Claim 6: We now solve for the steady-state distribution of this Markov chain. Let π_i and π'_i be the steady-state probabilities of states i for $i \geq -1$ and advantages states i' for all $i \geq 0$ respectively. The steady-state transition equations are given by

$$(1 - a - d)\pi_i = b(\pi_{i-1} + \pi'_i) + a\pi'_{i+1} \quad \text{for } i \geq 1, \quad (36)$$

$$(1 - d)\pi'_i = c(\pi_i + \pi'_{i+1}) \quad \text{for } i \geq 1, \quad (37)$$

$$(1 - c - d)\pi_{-1} = c(\pi_0 + \pi'_1), \quad (38)$$

$$(1 - a - d)\pi_0 = a\pi'_1 + (a + b)\pi_{-1}. \quad (39)$$

By rearranging the terms in (36)-(39), we get the following recurrence relation,

$$\pi_i = \frac{(1 - a - d)}{c}\pi_{i-1} - \frac{b}{c}\pi_{i-2} \quad \text{for } i \geq 2. \quad (40)$$

Solving the recurrence in (40) and simplifying (36)-(39) further, we can express π_i, π'_i for $i \geq 2$ in terms of π_1 as follows,

$$\frac{\pi_i}{\pi_{i-1}} = \frac{b}{c}, \quad (41)$$

$$\frac{\pi'_i}{\pi_i} = \frac{c}{a+c}. \quad (42)$$

From (41) we see that the Markov chain is positive-recurrent and a unique steady-state distribution if and only if $b < c$, which is equivalent to $p_1 < p_2$. If $p_1 \geq p_2$, the expected recurrence time to state 0, that is the time taken for U_2 to catch up with U_1 is infinity. When the Markov chain is positive recurrent, we can use (41) and (42) to solve for all the steady state probabilities. ■

Proof of Lemma 3: Since we always give priority to the primary user U_1 , we have $(\tau_1, \lambda_1) = (p_1, -\log(1-p_1))$. When $p_1 < p_2$, we can express the throughput τ_2 in terms of the steady state probabilities of the Markov chain in Fig 5. User U_2 experiences a throughput loss when it is in state -1 and the next slot is successful. Thus, when $p_2 > p_1$,

$$\tau_2 = p_2(1 - \pi_{-1}), \quad (43)$$

$$= p_2 \left(1 - \frac{c-b}{a+c} \right) = p_1. \quad (44)$$

If $p_1 \geq p_2$, the system drifts infinitely to the right side. There is a non-zero probability that in-order decoding via advantage states is not able to catch up and fill all gaps in decoding of U_2 . Thus, we cannot evaluate τ_2 using this Markov chain analysis.

To determine λ_2 , first observe that U_2 decodes an in-order packet when the system is in state 0 or states i' , for $i \geq 1$, and the next slot is successful. As given by Definition 6, the inter-delivery exponent λ_2 is the asymptotic decay rate of $\Pr(T_1 > t)$, the probability that no in-order packet is decoded by U_2 for t consecutive slots. To determine λ , we add an absorbing state F to the Markov chain as shown in Fig. 10, such that the system transitions to F when an in-order packet is decoded by U_2 .

In Fig. 10, all the states i and i' for $i \geq 1$ are fused into states I and I' because this does not affect the probability distribution of the time to reach the absorbing state F . The inter-delivery exponent λ_2 is equal to the rate of convergence of this Markov chain to its steady state, which is known to be (see [25, Chapter 4]) $\lambda_2 = -\log \xi_2$ where ξ_2 is the second largest eigenvalue of

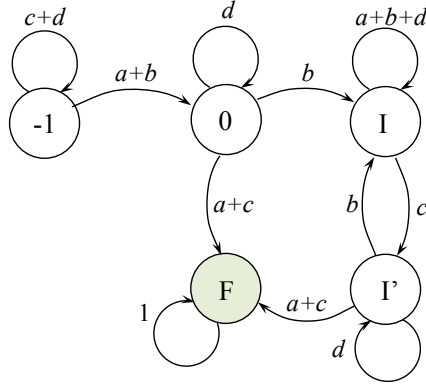


Fig. 10: Markov model used to determine the inter-delivery exponent λ_2 of user U_2 . The absorbing state F is reached when an in-order packet is decoded by U_2 . The exponent of the distribution of the time taken to reach this state is λ_2 .

the state transition matrix of the Markov chain,

$$A = \begin{pmatrix} d & b & 0 & 0 & a+c \\ 0 & a+b+d & c & 0 & 0 \\ 0 & b & d & 0 & a+c \\ a+b & 0 & 0 & c+d & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (45)$$

Solving for the second largest eigen-value of A , we can show that

$$\xi_2 = \max \left(1 - p_1, \frac{\left(1 - c + d + \sqrt{(1 - c + d)^2 + 4(bc + cd - d)} \right)}{2} \right). \quad (46)$$

Hence the inter-delivery exponent $\lambda_2 = -\log \xi_2$ is as given by (12). ■

Proof of Lemma 4: The state-transition equations of the Markov chain are as follows.

$$(\bar{d} - a)\pi_0 = a(\pi'_1 + \pi'_{-1}) + q_1(a + b)\pi_{-1} + q_2(a + c)\pi_1 \quad (47)$$

$$(\bar{d} - \bar{q}_2a - q_2b)\pi_i = q_2(a + c)\pi_{i+1} + \bar{q}_2b\pi_{i-1} + b\pi'_i + a\pi'_{i+1} \quad \text{for } i \geq 2 \quad (48)$$

$$(\bar{d} - \bar{q}_1a - q_1c)\pi_i = q_1(a + b)\pi_{i+1} + \bar{q}_1c\pi_{i+1} + c\pi'_i + a\pi'_{i-1} \quad \text{for } i \leq -2 \quad (49)$$

$$(\bar{d} - \bar{q}_2a - q_2b)\pi_1 = q_2(a + c)\pi_2 + b(\pi'_{-1} + \pi_0 + \pi'_1) + a\pi'_2 \quad (50)$$

$$(\bar{d} - \bar{q}_1a - q_1c)\pi_{-1} = q_1(a + b)\pi_{-2} + c(\pi'_{-1} + \pi_0 + \pi'_1) + a\pi'_{-2} \quad (51)$$

$$\bar{d}\pi'_i = \bar{q}_2c\pi_i + c\pi'_{i+1} \quad \text{for } i \geq 1 \quad (52)$$

$$\bar{d}\pi'_i = \bar{q}_2c\pi_i + b\pi'_{i+1} \quad \text{for } i \leq -1 \quad (53)$$

Rearranging the terms, we get the following recurrence in the steady-state probabilities on the right-side of the chain,

$$\alpha_3\pi_{i+3} + \alpha_2\pi_{i+2} + \alpha_1\pi_{i+1} + \alpha_0\pi_i = 0 \quad \text{for } i \geq 1 \quad (54)$$

where,

$$\alpha_3 = c(a + c)q_2 \quad (55)$$

$$\alpha_2 = -c\bar{d} + bcq_2 - (a + c)q_2\bar{d} \quad (56)$$

$$\alpha_1 = \bar{d}(\bar{d} - bq_2 - a\bar{q}_2) \quad (57)$$

$$\alpha_0 = -\bar{d}b\bar{q}_2 \quad (58)$$

The characteristic equation of this recurrence has the roots 1, ρ and ρ' . We can show that both ρ and ρ' are positive and at least one of them is greater than 1. The expression for the smaller root is,

$$\rho = -\frac{\alpha_3 + \alpha_2}{2\alpha_3} - \frac{\sqrt{(\alpha_3 + \alpha_2)^2 + 4\alpha_3\alpha_0}}{2\alpha_3} \quad (59)$$

The right-side of the Markov chain is stable if and only if $\rho < 1$. Thus, when $\rho < 1$, the steady-state probabilities π_i and π'_i for $i \geq 1$ are related by the recurrences,

$$\frac{\pi_{i+1}}{\pi_i} = \rho \quad \text{and} \quad \frac{\pi'_i}{\pi_i} = \frac{c(1 - q_2)}{\bar{d} - c\rho} \quad (60)$$

Similarly, for the left side of the chain we have the recurrences,

$$\frac{\pi_{i+1}}{\pi_i} = \mu \text{ and } \frac{\pi'_i}{\pi_i} = \frac{b(1 - q_1)}{\bar{d} - b\mu} \quad (61)$$

where

$$\mu = -\frac{\beta_3 + \beta_2}{2\beta_3} - \frac{\sqrt{(\beta_3 + \beta_2)^2 + 4\beta_3\beta_0}}{2\beta_3} \quad (62)$$

with the expressions β_k for $k = 0, 1, 2, 3$ being the same as α_k with b and c interchanged, and q_2 replaced by q_1 . We can use these recurrences we can express all steady-state probabilities π_i and π'_i for $i \geq 1$ in terms of π_1 , and the steady-state probabilities π_i and π'_i for $i \leq -1$ in terms of π_{-1} . Then using the states transition equation (47), and the fact that all the steady state probabilities sum to 1, we can solve for all the steady-state probabilities of the Markov chain.

User U_1 receives an innovative in every successful slot except when the source (with probability q_2) gives priority to U_2 in states $i, i \geq 1$. Thus, if $\rho < 1$ its throughput is given by

$$\tau_1 = p_1 \left(1 - q_2 \sum_{i=1}^{\infty} \pi_i \right) = p_1 \left(1 - \frac{q_2 \pi_1}{1 - \rho} \right) \quad (63)$$

Similarly if $\mu < 1$ we have,

$$\tau_2 = p_2 \left(1 - q_1 \sum_{i=-\infty}^{-1} \pi_i \right) = p_2 \left(1 - \frac{q_1 \pi_{-1}}{1 - \mu} \right) \quad (64)$$

Similar to the proof of Lemma 3, we determine the inter-delivery exponent λ_2 of user U_2 by adding an absorbing state F to the Markov chain as shown in Fig. 11, such that the system transitions to F when an in-order packet is decoded by U_2 . In Fig. 11, all the states i and i' for $i \geq 1$ are fused into states I and I' because this does not affect the probability distribution of the time to reach the absorbing state F . The inter-delivery exponent $\lambda_2 = -\log \xi_2$ where ξ_2 is the second largest eigenvalue of its state transition matrix of this Markov chain which is given by,

$$A = \begin{pmatrix} d & b & 0 & 0 & a + c \\ 0 & \bar{q}_2 a + b + d & c\bar{q}_2 & 0 & q_2(a + c) \\ 0 & b & d & 0 & a + c \\ q_1(a + b) & 0 & 0 & d + q_1 c + \bar{q}_1 b & \bar{q}_1(a + c) \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (65)$$

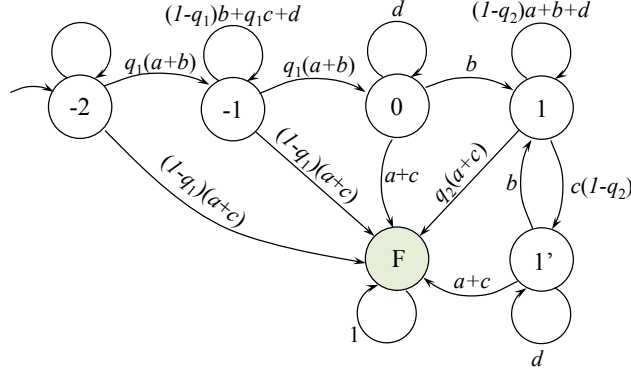


Fig. 11: Markov model used to determine the inter-delivery exponent λ_2 of user U_2 . The absorbing state F is reached when an in-order packet is decoded by U_2 . The exponent of the distribution of the time taken to reach this state is λ_2 .

Solving for the second largest eigen-value ξ_2 of A , we get

$$\xi_2 = \max \left(d + q_1 c + \bar{q}_1 b, \frac{\left(2d + \bar{q}_2 a + b + \sqrt{(2d + \bar{q}_2 a + b)^2 - 4(d(b + d) + \bar{q}_2(da - bc))} \right)}{2} \right). \quad (66)$$

The inter-delivery exponent $\lambda_2 = -\log \xi_2$ and is given by (14). The expression for the inter-delivery exponent λ_1 of user U_1 is same as (14) with b and c , and q_1 and q_2 interchanged. ■

REFERENCES

- [1] Sandvine Intelligent Networks, “Global Internet Phenomena Report,” <http://www.sandvine.com>, Mar. 2013.
- [2] Dropbox, <http://www.dropbox.com/>.
- [3] Google Docs, <http://docs.google.com/>.
- [4] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, “Practical loss-resilient codes,” in *ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1997, pp. 150–159.
- [5] E. Martinian, “Dynamic information and constraints in source and channel coding,” Ph.D. dissertation, MIT, Cambridge, USA, Sep. 2004.
- [6] A. Badr, A. Khisti, W. Tan and J. Apostoupoulos, “Robust Streaming Erasure Codes based on Deterministic Channel Approximations,” *International Symposium on Information Theory*, Jul. 2013.
- [7] P. Patil, A. Badr, A. Khisti and W. Tan, “Delay-Optimal Streaming Codes under Source-Channel Rate Mismatch,” *Asilomar*, Nov. 2013.

- [8] H. Yao, Y. Kochman and G. Wornell, "A Multi-Burst Transmission Strategy for Streaming over Blockage Channels with Long Feedback Delay," *IEEE Journal on Selected Areas in Communications*, Dec. 2011.
- [9] G. Joshi, Y. Kochman, G. Wornell, "On Playback Delay in Streaming Communication," *International Symp. on Information Theory*, Jul. 2012.
- [10] G. Joshi, "On playback delay in streaming communication," Masters thesis, Massachusetts Institute of Technology, 2012.
- [11] K. Mahadaviani, A. Khisti, G. Joshi, and G. Wornell, "Playback delay in on-demand streaming communication with feedback," *International Symposium on Information Theory (ISIT)*, Jul. 2015.
- [12] J. Cloud, D. J. Leith, and M. Médard, "A Coded Generalization of Selective Repeat ARQ," *IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2015.
- [13] M. Karzand, D. J. Leith, J. Cloud, and M. Médard, "Low delay random linear coding over a stream," *arXiv [cs.it] 1509.00167*, 2015.
- [14] L. Keller, E. Drinea and C. Fragouli, "Online Broadcasting with Network Coding," in *Network Coding Theory and Applications*, Jan. 2008, pp. 1–6.
- [15] J. Barros, R. Costa, D. Munaretto, and J. Widmer, "Effective Delay Control in Online Network Coding," in *International Conference on Computer Communications*, Apr. 2009, pp. 208–216.
- [16] A. Fu, P. Sadeghi, and M. Medard, "Delivery delay analysis of network coded wireless broadcast schemes," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, 2012, pp. 2236–2241.
- [17] J. Sundararajan, P. Sadeghi, and M. Médard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *IEEE Workshop on Network Coding, Theory, and Applications*, 2009, pp. 1–6.
- [18] J. Sundararajan, D. Shah and M. Médard, "Online network coding for optimal throughput and delay: the three-receiver case," in *International Symposium on Information Theory and its Applications*, Dec. 2008.
- [19] Y. E. Sagduyu and A. Ephremides, "On broadcast stability region in random access through network coding," in *44th Allerton Annual Conference on Communication, Control and Computing*, 2006, p. 38.
- [20] G. Joshi, Y. Kochman, G. Wornell, "The Effect of Block-Wise Feedback on the Throughput-Delay Trade-off in Streaming," *INFOCOM workshop on Communication and Networkin Techniques for Contemporary Video*, Apr. 2014.
- [21] A. Sahai, "Why Do Block Length and Delay Behave Differently if Feedback Is Present?" *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 1860–1886, May 2008.
- [22] G. Joshi, Y. Kochman, G. Wornell, "Throughput-Smoothness Trade-offs in Multicasting of an Ordered Packet Stream," *International Symposium on Network Coding*, Jun. 2014.
- [23] T. Cover and J. Thomas, *Elements of information theory*, 2nd ed. New York, NY, USA: Wiley-Interscience, 1991.
- [24] R. Durrett, *Probability: Theory and Examples*, 4th ed. Cambridge University Press, 2010.
- [25] R. Gallager, *Discrete Stochastic Processes*, 2nd ed. Kluwer Academic Publishers, 2013.